|  |
| --- |
|  |

|  |
| --- |
| **Architecture Design** |
| Human Resource Management Project |
|  |
| This document describes the architecture design for Human Resource Management (HRM) project. In this document, the view, including static view, physical view, and dynamic view will be shown. In addition, the data model will be described in this. |
|  |
| **KIM TUONG** |
| **[Pick the date]** |
|  |

Table of Contents

**No table of contents entries found.**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Documentation roadmap

# Scope

The purpose of this document is present the HRM system architecture under three perspectives. Three perspectives include:

* Static perspective
* Dynamic perspective
* Physical perspective

These perspectives are present base on the architectural drivers which have been identified in Architecture Design Document (ADD) document.

Besides, this document is present about quality attribute for HRM system. Two main quality attribute are:

* **Performance**:The response time of HRM system for each user interaction will be improved and the resource for each interaction will be reduced.
* **Security**: The HRM system will run safety. It cannot be accessed by unauthorized users, support detect the attacks and recover from the attack.
* **Usability:** The HRM system provide adequate user document including help, user manual and tutorials for user guidance.
* **Scalability:** The current system is just for about 10 users but it requires the system can expand more to 30- 40 users at a time.
* **Modifiability:** The HRM system supports the developers or maintainer can easy add new function or modify the current function whenever the business rules are change. The first release of HRM system just focuses on “Personal Information Management”. However, there will be more modules, which will be added to system in next release.
* **Availability**: The HRM will have periodically backup of database to ensure that whenever the database server is crashed, the data will not be lost.

# 1.2 Document organization:

This document includes three important parts:

**Part 1:** How to read this document

**Part 2:** System overview- The summarize of architectural drivers (High level requirement, quality attribute and constraint)

**Part 3:** View

*Physical* – How software and hardware interact with each other.

*Static* – The module of HRM system.

*Dynamic* – The interaction between HRM components.

The structure for presenting the view

**Session 1**: Primary presentation- The figure to present the view

**Session 2**: Element catalog – The table for describing the element which present in figure

**Session 3**: Element behavior- The flow of each component (This part is just for Dynamic perspective)

**Session 4:** Architecture background- Includes design decision and reasons for designing

1. System overview:

# 2.1 Project Overview

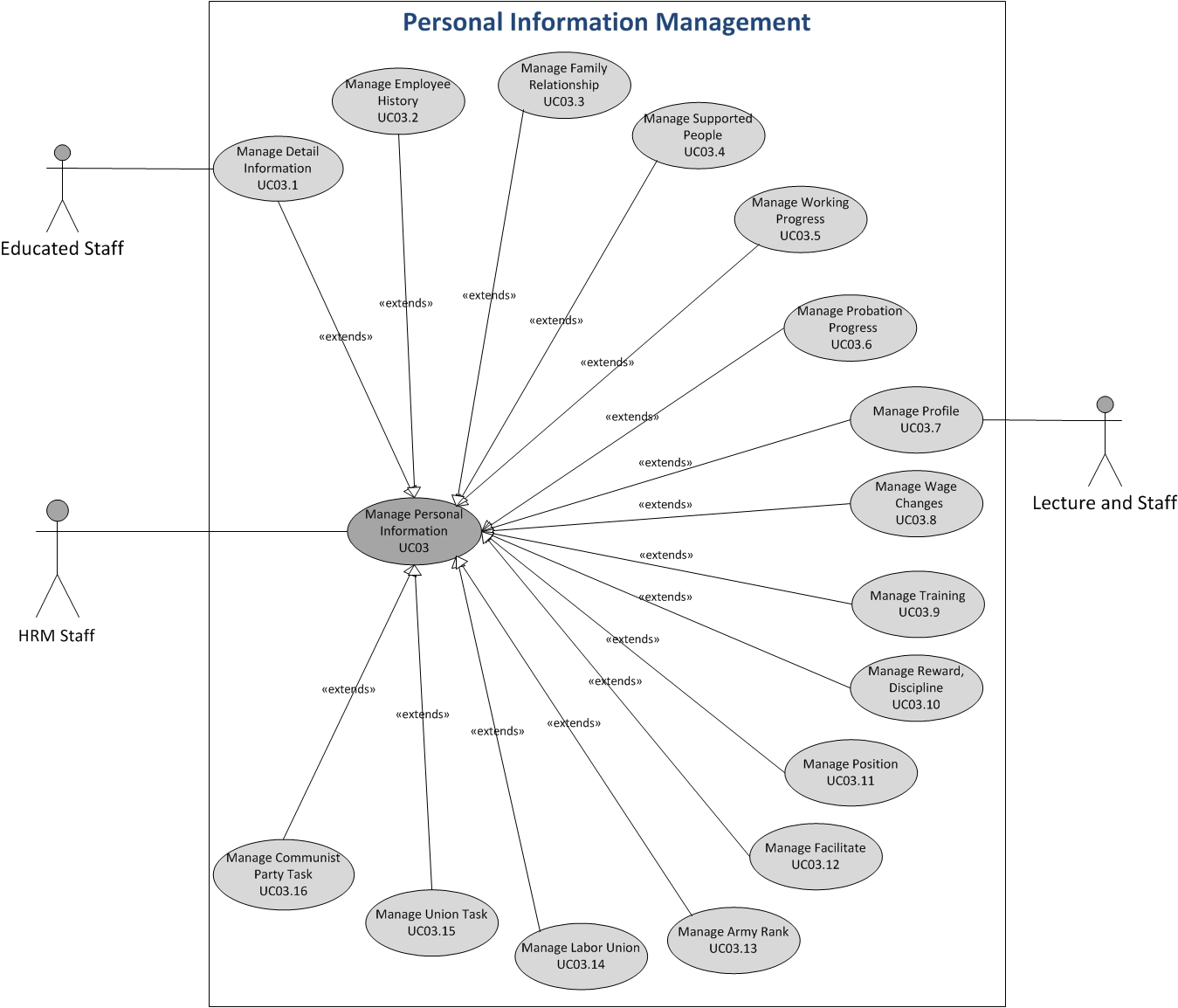
HRM is particularly developed for human resource management in university / colleges. The system consists of key modules:

* Personal information management
* Employee labor contract management
* Recruitment & training processing
* Payroll
* Administration panel - Utilities

# 2.2 Functional Requirement

Based on business rules and requirement, the list of functions in HRM system is listed in the following table. Based on the actor, we divided the function into 3 groups:

* Functions that related to administrator
* Functions that related to HRM Staff
* Functions that related to Lecture



*Figure 1: Use case of HRM project*

# 2.3 Quality Attribute Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System quality attributes** | | | **Priority** | **Concern** |  |
| **ID** | **Quality attributes** | **Description** | **Relation** |
| QA.01 | **PERFORMANCE** | -The time for staring the HRM system is fast.  -The response time of HRM system for each user interaction will be improved and the resource for each interaction will be reduced. |  | Response time |  |
| QA.02 | **SECURITY** | -The HRM system will run safety. It cannot be accessed by unauthorized users, support detect the attacks and recover from the attack.  -The data of human resources in Van Lang University will be protected from attacker. |  | Assign authority |  |
| QA.03 | **USABILITY** | -The HRM system have the consistent screens and easy to uses. It does not take time for training.  -The HRM system provide adequate user document including help, user manual and tutorials for user guidance |  | -Easy to use  -Show content |  |
| QA.04 | **SCALABILITY** | -The current system is just for about 10 users but it requires the system can expand more to 30- 40 users at a time. |  |  |  |
| QA.05 | **MODIFIABILITY** | -The HRM system supports the developers or maintainer can easy add new function or modify the current function whenever the business rules are change. The first release of HRM system just focuses on “Personal Information Management”. However, there will be more modules, which will be added to system in next release. |  | Add/remove/modify functions |  |
| QA.06 | **AVAILABILITY** | -The HRM will have periodically backup of database to ensure that whenever the database server is crashed, the data will not be lost. |  | Recover and repair the faults |  |

1. System overview:

# Technical Constraints

|  |  |  |
| --- | --- | --- |
| **ID** | **Constraint name** | **Constraint Description** |
| TC.PIM.1 | Database | The system database will be developed using SQL server |
| TC.PIM.2 | Development framework | .Net 4.0, WCF, Silverlight, Entity framework |
| TC.PIM.3 | Network | Network is ADSL/Mega WAN |
| TC.PIM.4 | Programming language | Using C#, program and fix code on XML file or properties of XML file |
| TC.PIM.5 | Language | Default language is Vietnamese |
| TC.PIM.6 | Third-party | - Using Microsoft Word, Excel for documenting, importing, and exporting the data  -Using Telerik to design the interface |

# 3.2 Business Constraints

|  |  |  |
| --- | --- | --- |
| **ID** | **Constraint name** | **Constraint description** |
| BC.PIM.7 | Time limitation | The end of the project is in April 31st,2012 |

1. System Context:

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Educated Department | * Manage issues related educated, papers educated. * Manage schedule teach hour, student, student transcript |
| Administrator | * Manage system, manage information relate with staff of HRM (besides may be have staff, lecture) |
| Salary Group | * Manage salary issue for school |
| Department | * Manage recruitment issues, hour work, training staff & lectures. |
| HR Group | * Manage human resource issues. |
| Account Department | * Manage accountant (salary for staff, lecture). |



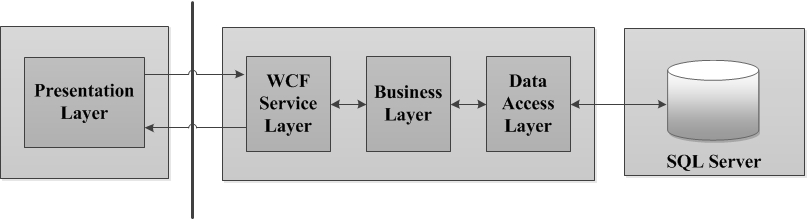
*Figure 2: System Context of HRM project*

1. View:

# 5.1 Static Perspective

# 5.1.1 Decompose in first level of HRM system

# 5.1.1.1 Primary Presentation



*Figure 3: First Decomposition of HRM system*

# 5.1.1.2 Element Catalog:

|  |  |  |
| --- | --- | --- |
| **Element** | **Type** | **Description** |
| **Presentation Layer** | Layer | This layer will contain the modules that related the interface (Silverlight). These will support for developing web component. In addition, this layer also contains MVVM (Model-View-ViewModel) object, data, and common. |
| **WCF Service Layer** | Layer | This layer will contain the security service. |
| **Business Layer** | Layer | This layer will contain the modules for the business implementing, including Application Façade, Common service, and Business Service |
| **Data access Layer** | Layer | This layer will contain the modules for mapping between Business layer and database, contains the Data Access object and entity framework. |
| **SQL Server** | Database | It is responsible for all object storage and database operation. |
| **Allow to use** | Relationship | Upper layers can use lower layers and reversly, lower layers can use layers above. |

# 5.1.1.3 Architecture background:

# 5.1.1.3.1 Design decision:

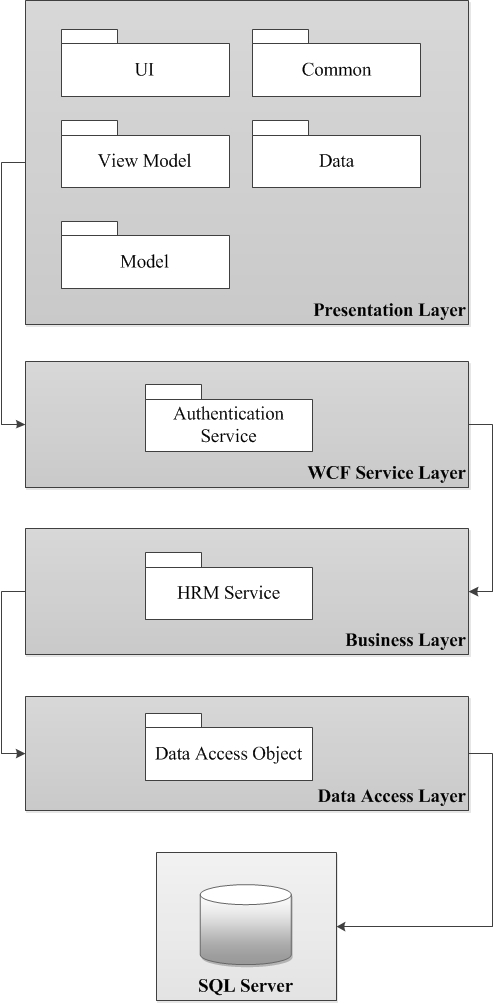
- Use n-tier model for architect, including presentation, service, business, and data access tier. It allows us to create a flexible and reusable application, easily modify or add a specific layer, rather than have to rewrite the entire application over

# 5.1.1.3.2 Design rationale:

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Modifiability | QA.07 | Easy to change |  |
| Rationale:   * Using MVVM model, it allows us changing the GUI of application but not to impact to code behind, easily to test, maintain, and develop. | | |

# 5.1.2 Decompose in second level of HRM system

# 5.1.2.1 Primary Presentation



# 5.1.2.2 Element Catalog

|  |  |  |
| --- | --- | --- |
| **Element** | **Type** | **Description** |
| **UI** | Module | This module represents the user interface of the application for HRM staff. |
| **Common** | Module | Contains the many classes for common using between the layers |
| **Data** | Module | Using for connecting to Entity Framework in Database |
| **View Model** | Module | This module reference the objects that are necessary for UI from presentation logic and data |
| **Model** | Module | This module packages the business logic and data. It is responsible for managing the data of application and ensure the it’s consensus |
| **Authentication Service** | Module | Enables us to authenticate users through a Windows Communication Foundation (WCF) service. Through the Authentication Service class, you can log users in, log users out, validate credentials, check authentication status, customize authentication, and set the authentication cookie. |
| **HRM Service** | Module | Business process objects allow us to implement HRM business rules and provide transaction support if required. |
| **Data Access object** | Module | Allow for connecting between Service layer and database. |

# 5.1.2.3 Architecture background

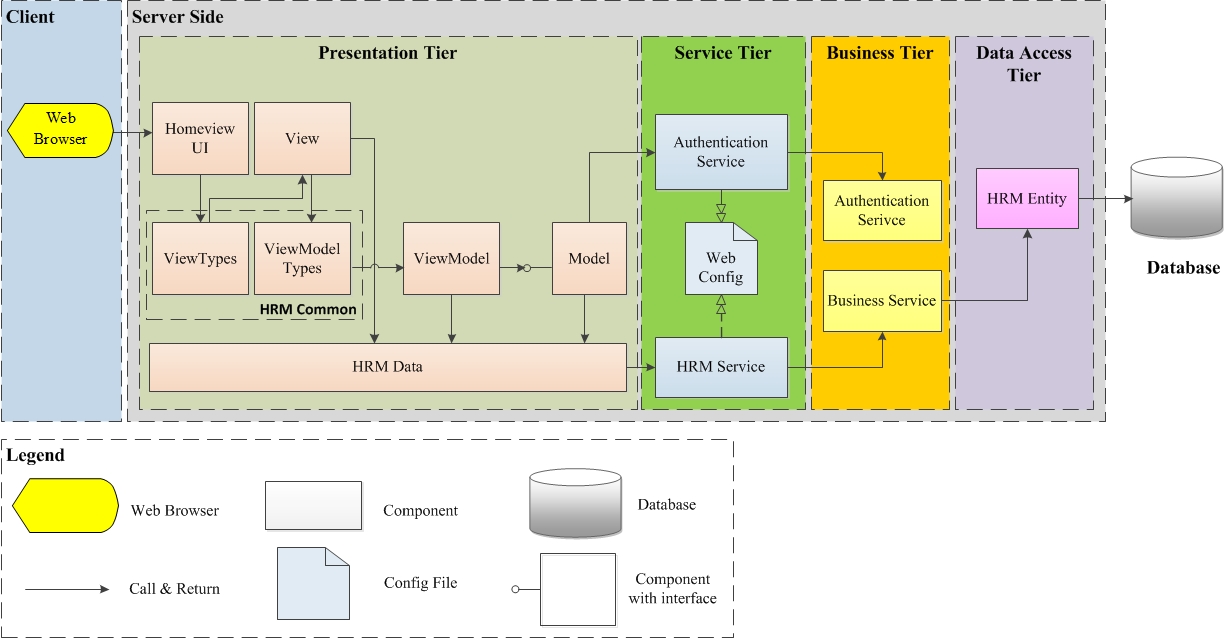
# 5.1.2.3.1 Design decision

- Using Model-View-ViewModel (MVVM) pattern. MVVM pattern separate the business layer and the presentation from the user interface (UI). It means that it allows us changing the GUI of application but not to impact to code behind.

# 5.1.2.3.2 Design rationale

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA.02 | Response time | The response time for updating and modifying the “Personal Information” will be less than 4 seconds and the number of transaction are 10. |
| Rationale   * Using tactic “Maintain multiple copies of either data or computations”. The PIM system will use WCF service, so the performance will be increased because it can handle the large number of user transactions. * Entity framework is also promoting performance. It’s likely an object/relational mapping framework that allow the business layer working directly with objects without through database. | | |
| Security | QA.08 |  |  |
| Rationale:   * The WCF supports for exchanging the secure message based on security foundation, e.g. https, windows integrated security, or access control (use username and password for authentication) | | |

# 5.2 Dynamic Perspective



# 5.2.1 Element Catalog

|  |  |  |
| --- | --- | --- |
| **Element** | **Type** | **Description** |
| **Web browser** | Component | This component represents the user interface of the application running on a web browser. The client will enter the address and start the application on Web |
| **Homeview UI** | Component |  |
| **HRM data** | Component |  |
| **ViewTypes** | Component |  |
| **ViewModel Types** | Component |  |
| **Web Config** |  |  |
| **HRM service** |  |  |
| **Business Service** |  |  |
| **HRM entity** |  |  |

# 5.2.2 Design decision

- Using Model-View-ViewModel (MVVM) pattern. MVVM pattern separate the business layer and the presentation from the user interface (UI). It means that it allows us changing the GUI of application but not to impact to code behind.

- Using WCF in Service tier. The WCF supports for exchanging the secure message based on security foundation, e.g. https, windows integrated security, or access control (use username and password for authentication)

- Using n-tier model for architect, including presentation, service, business, and data access tier. It allows us to create a flexible and reusable application, easily modify or add a specific layer, rather than have to rewrite the entire application over

- Using the Entity framework that allows the business layer working directly with objects without through database.

# 5.2.3 Design Rationale

# 5.2.4 Behavior diagram

